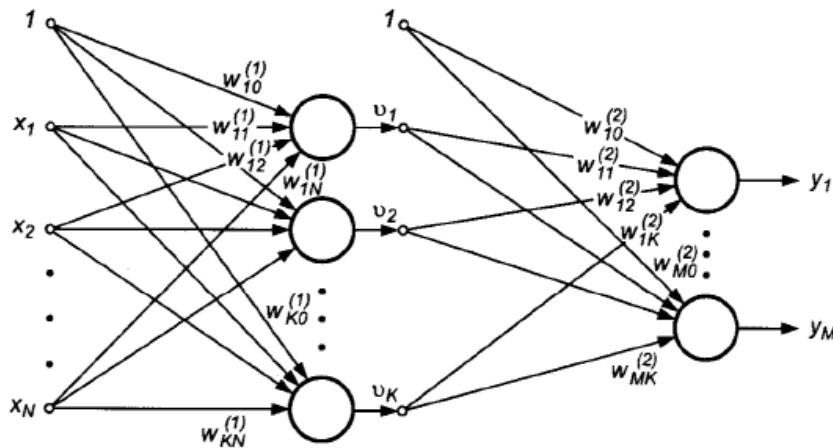


„Podstawy Sztucznej Inteligencji”

Scenariusz 3

Temat ćwiczenia: *Budowa i działanie sieci wielowarstwowej typu feedforward*

1. Sieć wielowarstwowa – sieć złożona z neuronów ułożonych w wielu warstwach przy czym oprócz warstwy wejściowej i wyjściowej istnieje co najmniej jedna warstwa ukryta. Jest to przykład sieci feedforward, w której jednoznacznie zdefiniowany jest kierunek przepływu impulsów. Perceptrony pogrupowane są w warstwy, dane wejściowe wchodzi do wszystkich jednostek w najniższej warstwie.



Rys. 3.5. Ogólny schemat sieci neuronowej sigmoidalnej dwuwarstwowej (o jednej warstwie ukrytej)

Algorytm wstecznej propagacji – określa strategię doboru wag w sieci wielowarstwowej przy wykorzystaniu gradientowych metod optymalizacji. Uważany za jeden ze skuteczniejszych algorytmów uczenia sieci wielowarstwowej. Podstawę algorytmu stanowi funkcja celu, definiowana zwykle jako suma kwadratów różnic między aktualnymi wartościami sygnałów wyjściowych sieci, a wartościami zadanymi.

2. Listing programu

```
close all; clear all; clc;

dane_wej = [-2.0 -1.8 -1.6 -1.4 -1.2 -1.0 -0.8 -0.6 -0.4 -0.2 0
0.2 0.4 0.6 0.8 1.0 1.2 1.4 1.6 1.8 2.0];
dane_wyj = [1.6633e+03 1.6901e+03 1.6880e+03 1.6741e+03 1.6867e+03 1.7329e+03 1.7764e+03 1.7860e+03
1.7800e+03 1.8036e+03 0 1.9289e+03 1.9495e+03 1.9490e+03 1.9825e+03 2.0655e+03
2.1477e+03 2.1784e+03 2.1791e+03 2.2203e+03 2.3262e+03];

test = zeros(1);
net = feedforwardnet(4); %wielowarstwowa siec
net.trainFcn = 'traingd'; %algorytm wstecznej propagacji
net.trainParam.lr = 0.5; %wspolczynnik uczenia
net.trainParam.mc = 0; %wspolczynnik bezwladnosci
net = train(net, dane_wej, dane_wyj); %trenowanie

wyniki = zeros(size(net));
```

```

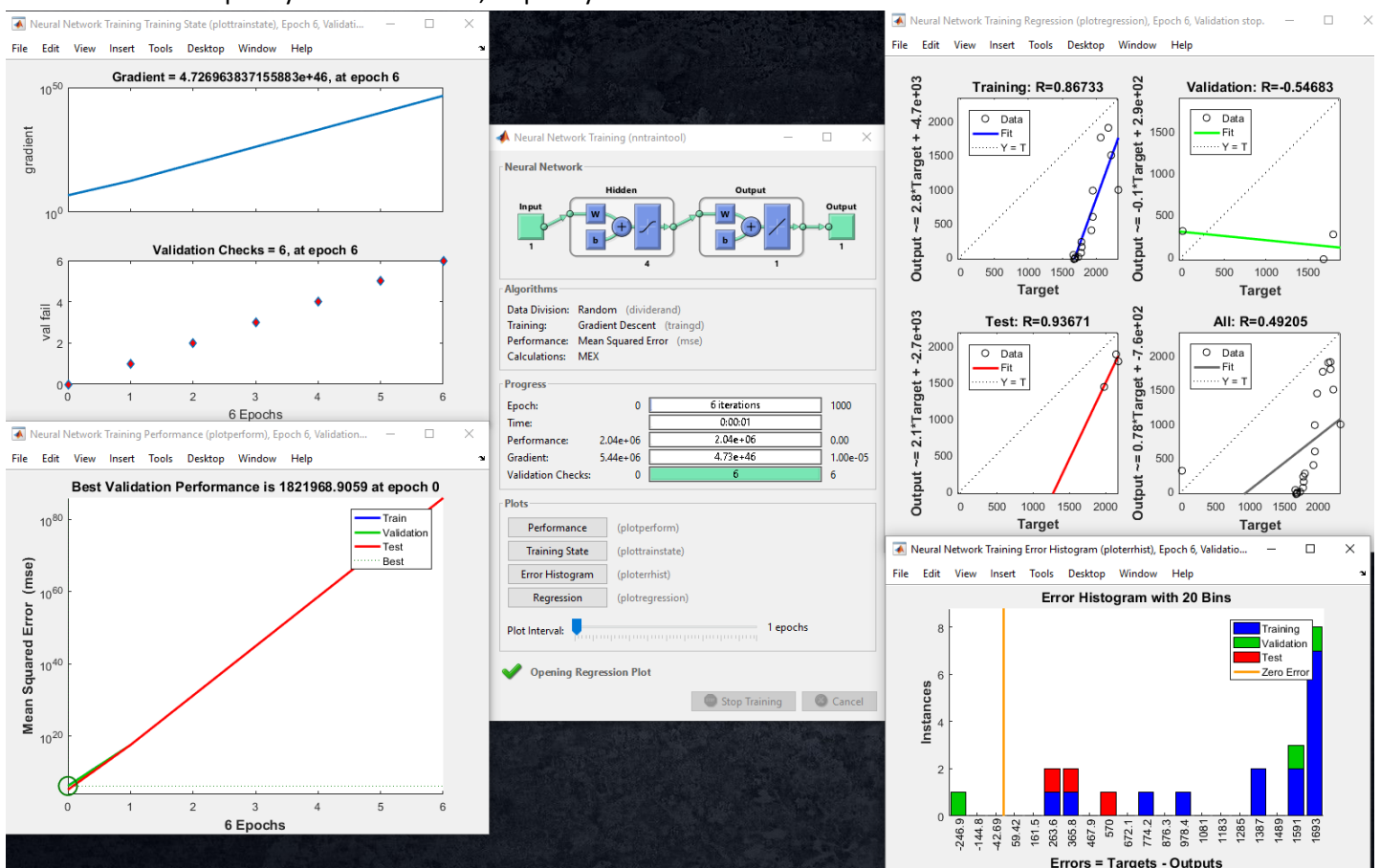
function w =rastrigin(x)
    if x==0
        w=0;
    else
        x1=x;
        A=10;
        n=100;
        dx=(5.12-x)/n;
        w=A*n;

        for i=1:1:n
            x=x1+(i*dx);
            w=w+x^2-A*cos(2*pi*x);
        end
    end
end

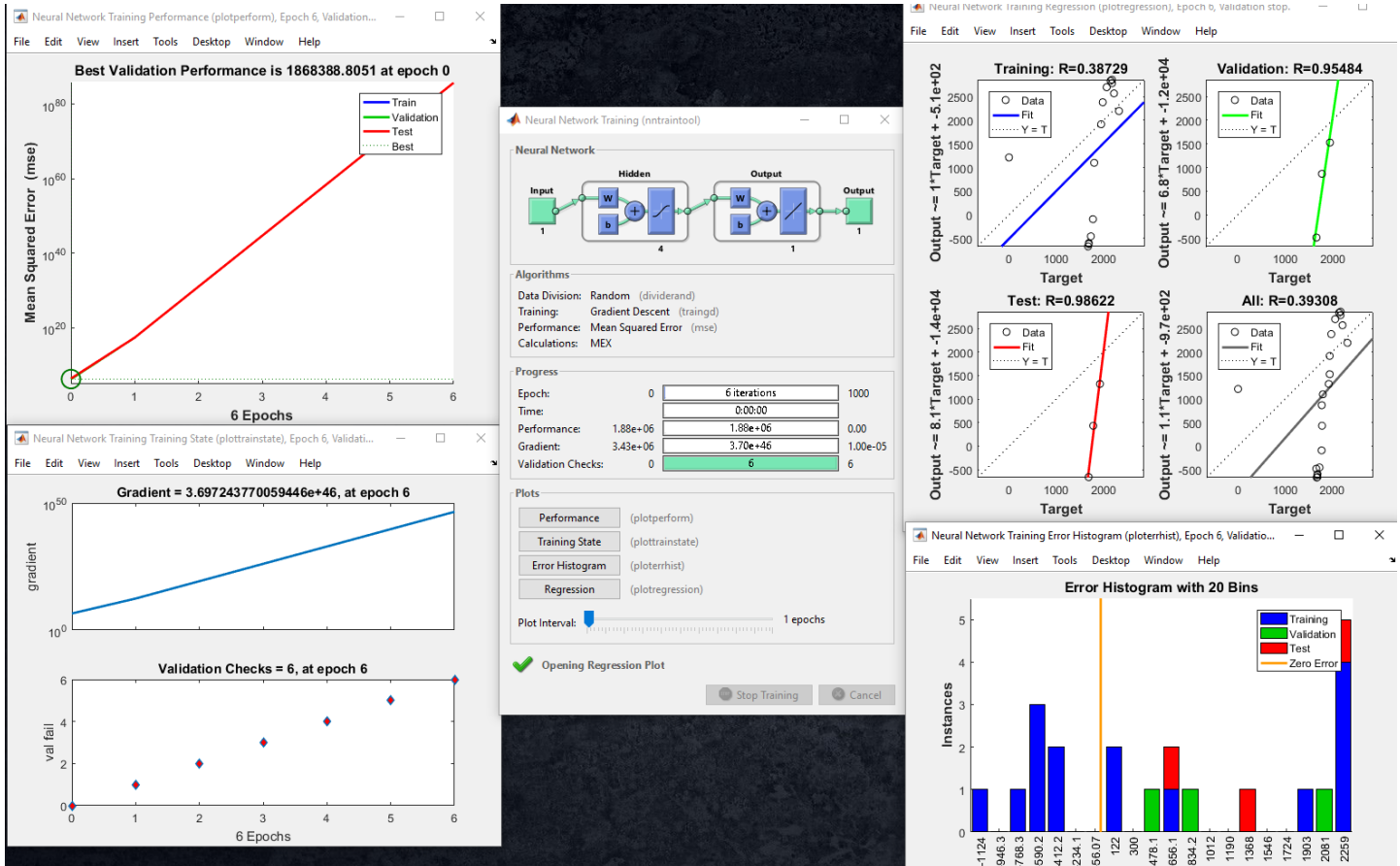
```

- Wygenerowane zostały dane uczące i testujące dla funkcji rastrigin dla danych wejściowych z przedziału $[-2,2]$. Kolejnym krokiem było uczenie sieci dla różnych współczynników uczenia i bezwładności oraz testowanie sieci. Wyniki działania programu znajdują się poniżej.

Współczynnik uczenia 0.5, współczynnik bezwładności 0



Współczynnik uczenia 0.5, współczynnik bezwładności 0.5



4. Wnioski

- współczynniki bezwładności i uczenia mocno wpływają na wyniki
- im większy współczynnik bezwładności tym większe błędy
- najmniej dokładne wyniki otrzymuje się, gdy oba współczynniki są równe
- współczynniki nie mają wpływu na wartość epoch